ECE3073
Looking at example executions

Clive Maynard
2020

Two approaches

- Run some uCOS II code under windows and observe the text output from the task executions.
- Use a scheduling simulator to observe the expected execution as a timeline.

Example executions

- Example 1:
- Two timed tasks with specified priorities

Task 1

```
//*********************************************************************************************
// TASK:
//              Task1
// DESCRIPTION:
//              Prints a message with elapsed timer ticks and goes to sleep for 500 msec
//*********************************************************************************************
void  Task1 (void *p_arg){
          p_arg = p_arg;
              while (1){
                    OS_Printf("Task1:\t\t Ticks=%d,\t delaying 500 msec\n", OSTimeGet());
                    /* your code here. Create more tasks, etc.                          */
                    OSTimeDlyHMSM(0, 0, 0, 500);

              }
} //Task1
```

What does the "Task 1" do?

- Runs an infinite loop with a time delay and print.

```
//*********************************************************************************************************
// TASK:
//                      StartTask
// DESCRIPTION:
//                      First task created in app.c.
//                      Creates another task with error checking and then enters an infinite loop.  Each loop iteration has a blocking call to OS.
//                      In this case it is OSTimeDlyHMSM.
//                      Prints a message with elapsed timer ticks and goes to sleep for 1 sec
//*********************************************************************************************************

void  StartTask (void *p_arg)
{
        INT8U  err;
   p_arg = p_arg;  // removes compiler warning of unused p_arg

#if OS_TASK_STAT_EN > 0
   OSStatInit();                        /* Determine CPU capacity                                               */
#endif
        OS_Printf("StartTask:\t %s VERSION %d\n", __FILENAME__, VERSION);
        OS_Printf("StartTask:\t Creating Task1 with priority %d\n", TASK1_PRIO);
                err = OSTaskCreateExt(Task1,
            (void *)0,
            (OS_STK *)&Task1Stk[TASK_STK_SIZE-1],
            TASK1_PRIO,
            TASK1_PRIO,
            (OS_STK *)&Task1Stk[0],
            TASK_STK_SIZE,
            (void *)0,
            OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR);
        Perror(err,"ERROR - OSTaskCreate(Task1 ..) failed"); // checks err and exits if an error printing message - see app.c
        #endif

   while (1)                        /* Task body, always written as an infinite loop.  */
        {
                OS_Printf("StartTask:\t Ticks=%d,\t delaying 1 second\n", OSTimeGet());
           OSTimeDlyHMSM(0, 0, 1, 0);
   }
} //StartTask
```

What does the "Start Task" do?

- Creates Task 1 then
- Runs an infinite loop with a time delay and print.

What makes them distinct?

- The time delays and the priorities allocated
- Task 1 ……..500msecs
- StartTask ……..1sec
- for version 1 …….same priority!!
- for version 2 …….Task 1 a higher number
- for version 3 …….Task 2 a lower number

Priorities in uCOS II

- The lowest number represents the highest priority
- 0 is the highest and 64 the lowest.
- What priority should we give to the "Idle Task"
- **Always check when looking at a new system!!!**
- The simulator has the highest number as the highest priority.

Predictions for Version 1??

main: Creating StartTask with priority 5

StartTask: example1_tasks.c VERSION 1

StartTask: Creating Task1 with priority 5

Task1: Ticks=?, delaying 500 msec

StartTask: Ticks=?, delaying 1 second

Run it and determine the problem!

Predictions for Version 2?

main: Creating StartTask with priority 5

StartTask: example1_tasks.c VERSION 2

StartTask: Creating Task1 with priority 6

Task1: Ticks=?, delaying 500 msec

StartTask: Ticks=?, delaying 1 second

Task1: Ticks=?, delaying 500 msec

StartTask: Ticks=?, delaying 1 second

The Result for eg1v2

```
Z:\Desktop\Ex1OS\SoftwareForStudents\RealTimeExamplesPDFsEXEs>eg1v2.exe
main:          Creating StartTask with priority 5
StartTask:      example1_tasks.c VERSION 2
StartTask:      Creating Task1 with priority 6
StartTask:      Ticks=1,       delaying 1 second
Task1:         Ticks=1,       delaying 500 msec
Task1:         Ticks=51,      delaying 500 msec
StartTask:      Ticks=101,     delaying 1 second
Task1:         Ticks=101,     delaying 500 msec
Task1:         Ticks=151,     delaying 500 msec
StartTask:      Ticks=201,     delaying 1 second
Task1:         Ticks=201,     delaying 500 msec
Task1:         Ticks=251,     delaying 500 msec
StartTask:      Ticks=301,     delaying 1 second
Task1:         Ticks=301,     delaying 500 msec
Task1:         Ticks=351,     delaying 500 msec
StartTask:      Ticks=401,     delaying 1 second
Task1:         Ticks=401,     delaying 500 msec
^C
```

SimSo

- Two versions of SimSo
- Download and execute
- Web based
- We will set up the example to mimic what we have just done.

The configuration

Two tasks with StartTask having the higher priority

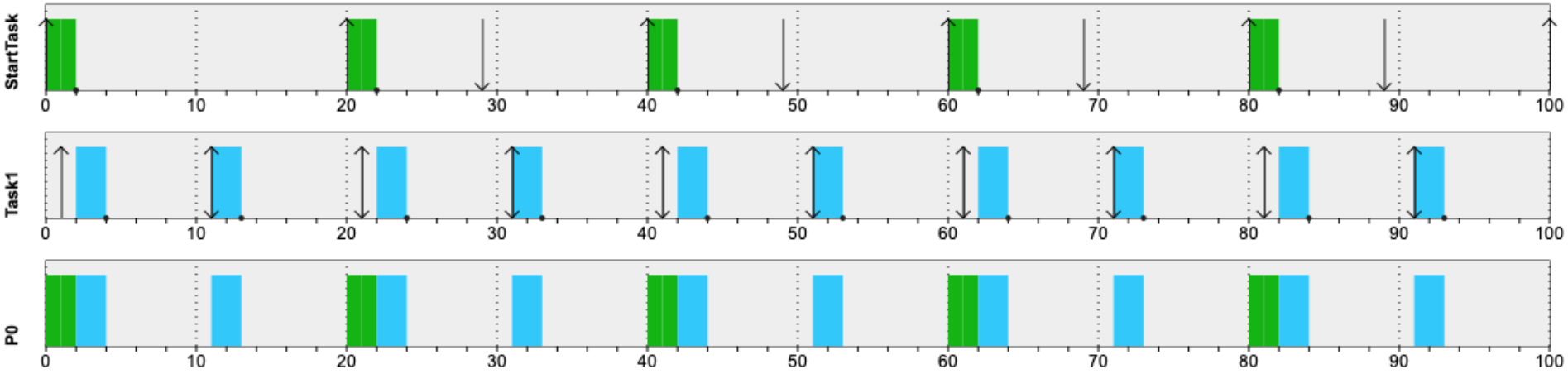
Task 1 is activated within StartTask so is slightly later to start.

StartTask has twice the delay of Task 1

1 Processor and fixed priority.

Deadlines same as period

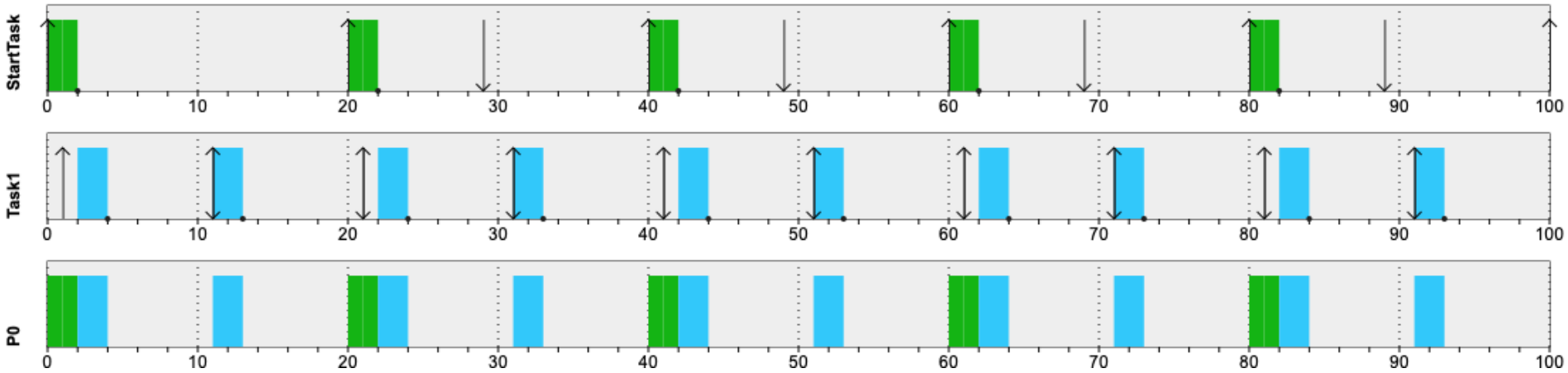
The output from Simso



Are we happy?

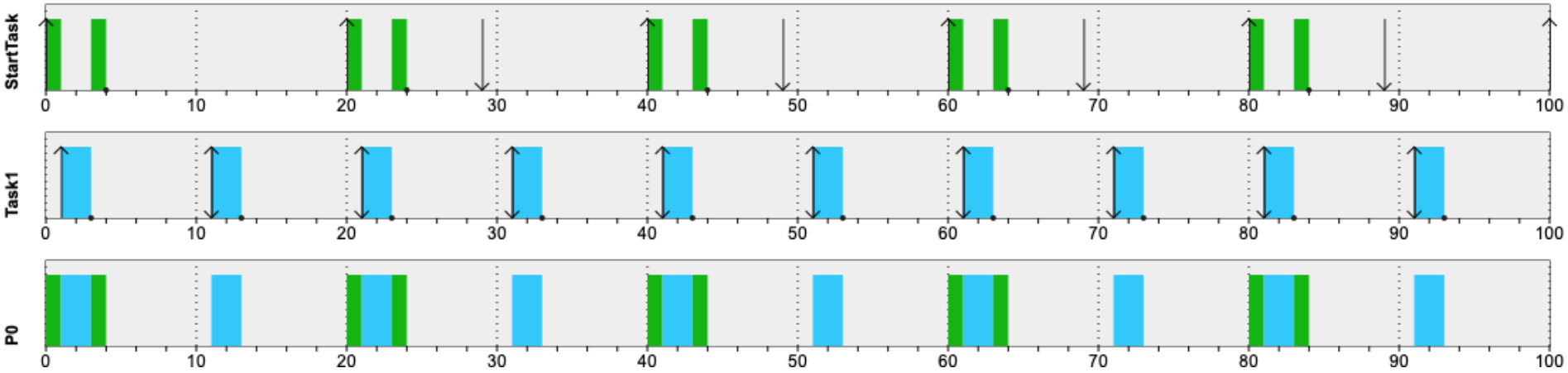
Where did we include our priority settings?

Adding StartTask priority 10 and Task1 priority 5



No change!!

What if we swapped priorities?



Predictions for Version 3?

main: Creating StartTask with priority 5

StartTask: example1_tasks.c VERSION 3

StartTask: Creating Task1 with priority 4

Task1: Ticks=?, delaying 500 msec

StartTask: Ticks=?, delaying 1 second

Task1: Ticks=?, delaying 500 msec

StartTask: Ticks=?, delaying 1 second

Task1: Ticks=?, delaying 500 second

Alternate Predictions for Version3

main: Creating StartTask with priority 5

StartTask: example1_tasks.c VERSION 3

StartTask: Creating Task1 with priority 4

Task1: Ticks=?, delaying 500 msec

StartTask: Ticks=?, delaying 1 second

Task1: Ticks=?, delaying 500 msec

Task1: Ticks=?, delaying 500 msec

StartTask: Ticks=?, delaying 1 second

Where do we go from here?

- SimSo is good for scheduling simulation.
  So we will use it for this.
- It does not handle inter task communications.
- BUT that comes a little later.

Example executions

- Example 2:
- Two timed tasks with parameter passing

Predictions ex 2

main: Creating StartTask with priority 5

StartTask: example2_parameterpassing.c VERSION 1

StartTask: Creating Task with priority 6 and parameter 0

StartTask: Creating Task with priority 7 and parameter 1

TaskStart: Ticks=?, still here!

Task0: Ticks=?, still here

Task1: Ticks=?, still here

Task0: Ticks=?, still here

Task1: Ticks=?, still here

Predictions ex 2

main: Creating StartTask with priority 5

StartTask: example2_parameterpassing.c VERSION 2

StartTask: Creating Task with priority 6 and parameter 0

StartTask: Creating Task with priority 7 and parameter 1

TaskStart: Ticks=?, still here!

Task0: Ticks=?, still here

Task1: Ticks=?, still here

Task0: Ticks=?, still here

Task1: Ticks=?, still here

Predictions ex 2

main: Creating StartTask with priority 5

StartTask: example2_parameterpassing.c VERSION 3

StartTask: Creating Task with priority 4 and parameter 0

StartTask: Creating Task with priority 7 and parameter 1

TaskStart: Ticks=?, still here!

Task0: Ticks=?, still here

Task1: Ticks=?, still here

Task0: Ticks=?, still here

Task1: Ticks=?, still here

Predictions ex 2

main: Creating StartTask with priority 5

StartTask: example2_parameterpassing.c VERSION 4

StartTask: Creating Task0 with priority 6 and parameter 0

Task0: Ticks=?, still here

StartTask: Creating Task1 with priority 7 and parameter 1

Task1: Ticks=?, still here

StartTask: Creating Task2 with priority 8 and parameter 2

Task2: Ticks=?, still here

TaskStart: Ticks=?, still here!

Task0: Ticks=?, still here

Task1: Ticks=?, still here

Task2: Ticks=?, still here